Содержание

Предисловие

Услышав, что мы приступаем к работе над Dart – структурным, масштабируемым языком с быстрой виртуальной машиной, развитым редактором и компилятором на JavaScript, я поначалу не поверил. «Быть может, именно этот проект сделает веб-программирование проще для таких разработчиков, как я сам?» – задавался я вопросом в надежде получить утвердительный ответ. Имея опыт работы со структурными языками и мощными инструментальными средствами разработки, я ожидал какого-нибудь более продуктивного способа создавать современные крупномасштабные веб-приложения. Проект Dart казался как раз тем, что я искал. Я вырос на таких объектно-ориентированных языках, как C++, Java и Ruby, которые применял при построении своих первых интерактивных веб-сайтов, а позже обогащенных веб-приложений с развитой клиентской частью. Я научился продуктивно работать с классами, объектами и модульным кодом. Я ценил интегрированные среды разработки (IDE) за встроенные в них средства анализа, рефакторинга и навигации, потому что они помогали мне писать большие и сложные приложения. Жизнь была прекрасна. Я искал новые направления приложения своих знаний, и мне повезло получить работу в команде, работавшей над браузером Chrome. Поначалу я осваивал современный браузер и с воодушевлением принялся изучать многочисленные новые возможности HTML5. Ныне сеть веб развивается так быстро и у нее появилось столько пользователей, что находиться в центре событий безумно интересно. Жизнь стала еще прекрасней.

И хотя итеративный процесс веб-разработки с его коротким циклом мне очень нравился, все же не хватало столь любимых мной структурных языков и полезных инструментов. Я хотел, чтобы программы для современных браузеров можно было писать в IDE, умеющих выполнять автозавершение кода, на языках с настоящими классами... перечислять можно долго.

Поэтому, услышав о Dart, я ухватился за представившуюся возможность. Писать для самой интересной платформы, да еще не от-

казываясь от навыков и инструментов, с которыми хорошо знаком и умеешь работать? Да кто же будет против!

И я был не единственным разработчиком, кто с радостью присоединился к проекту. Автор этой книги Крис Бакетт был одним из первых, кто принял Dart. Он завел свой блог Dartwatch в тот самый день, как Google анонсировала Dart, и этот блок по сей день живет и процветает. Крис участвовал в проекте с самого начала, поэтому естественно, что именно он написал одну из первых книг, призванных помочь другим разработчикам в изучении Dart.

Крис – потрясающий автор, потому что сумел написать книгу в условиях, когда язык и его библиотеки претерпевали постоянные изменения. Ему удалось осветить многочисленные аспекты проекта Dart. Особенно мне понравились примеры, в которых иллюстрируется не только функциональность базового языка, но и более сложные вопросы, касающиеся HTML5. Крис рассматривает одностраничные приложения и показывает, как с помощью Dart создаются современные приложения, работающие в браузере. Но и это не все – вы научитесь программировать на Dart даже серверные приложения!

После года напряженной работы, десятков тысяч сохранений в системе управления версиями, исправления тысяч ошибок и непрерывающейся обратной связи с сообществом мечта о структурном языке для веб-программирования стала реальностью. И хотя разработка Dart еще не завершена, сегодня мы – благодаря книге Криса – уже можем создавать замечательные приложения для современного Интернета. Ликуйте!

Сет Лэдд,
разработчик и пропагандист
Google

Вступление

В октябре 2011 года подтвердились слухи о том, что Google разработала новый язык, ориентированный на разработку сложных веб-приложений характерного для Google масштаба. Спустя месяц в сети был обнародован внутренний документ Google под названием «Future of JavaScript» (Будущее JavaScript), из которого следовало, что в Google ведутся работы над языком, более подходящим для веб-разработки, чем JavaScript. Упоминалось даже предположительное название языка – Dash. Идея о создании нового языка родилась из-за медленного развития JavaScript, обусловленного отчасти тем, что в нем заинтересовано слишком много сторон и комитетов. Целью же было показать, каким мог бы стать JavaScript, если бы он был придуман сегодня. Основная задача формулировалась так: «сохранить динамическую природу JavaScript, но повысить производительность и обеспечить поддержку со стороны инструментальных средств при разработке крупных проектов». Кроме того, язык должен был допускать кросс-компиляцию на JavaScript. Апробационная техническая версия языка была представлена миру и получила название Dart.

Я как раз тогда закончил работу над крупным заказным справочным приложением, написанным на GWT и предназначенным для развертывания на устройствах с сенсорными экранами в среде, неблагоприятной для компьютеров. Технология Google Web Toolkit (GWT) была разработана Google для кросс-компиляции Java на JavaScript. GWT позволяет разработчику воспользоваться модульной структурой, типобезопасностью и инструментальной поддержкой Java при написании ориентированных на браузеры программ, не требующих подключения дополнительных модулей, таких как Flash или Silverlight. Потратив предшествующие два года на написание GWT-кода и координацию работы программистов из трех стран, я понимал, насколько ценны средства проверки кода в точках интеграции – то, чего так не хватает языку JavaScript. Меня также привлекала возможность использовать один и тот же код на сто-

роне клиента и сервера – я видел, какие из этого можно извлечь преимущества.

Горя желанием узнать как можно больше о новом языке Dart, я прочитал всю имевшуюся документацию. На тот момент она состояла из исходного кода, нескольких примеров и спецификации языка. Мне казалось, что раз уж я потратил время на то, чтобы уложить эти знания у себя в голове, то было бы хорошо поделиться ими с публикой с помощью блога. Тогда-то я и завел блог Dartwatch, где опубликовал серию статей о том, как решаются типичные задачи в Dart: организация проекта, создание классов, взаимодействие с браузером. Одно цепляло за собой другое – и в конце концов издательство Manning обратилось ко мне с предложением написать книгу о Dart. Через год результат этой работы вышел из печати.

За минувший год у Dart было время созреть, а его разработчики внимательно прислушивались к отзывам и реагировали на них. Близилось время первой контрольной точки Dart, и в первоначальную спецификацию языка было внесено множество изменений по предложениям первых приверженцев, пытавшихся применить его для решения реальных задач. Сообщество этих приверженцев также занималось разработкой различных инструментов и библиотек, в том числе драйверов баз данных, библиотек двумерной и трехмерной графики и каркасов. Многие из них можно найти на сайте GitHub или Dartwatch.

Первая контрольная точка Dart (Milestone 1) – это крупное достижение, дающее разработчикам возможность использовать базовый язык для создания впечатляющих библиотек и API, которые помогут превратить Dart в язык «все включено», каким его и видят идеологи в Google. С каждым днем Dart становится все лучше, а благодаря открытости проекта вы можете наблюдать за тем, что многочисленные разработчики помещают в репозиторий исходного кода (и даже принять в этом участие). Надеюсь, что эта книга поможет вам создать замечательные приложения на Dart.

Благодарности

Выяснилось, что написание книги – вовсе не такое простое дело, как мне казалось поначалу. Не будь поддержки со стороны всех участвующих в проекте сотрудников издательства Manning, вы вряд ли читали бы ее сегодня. Спасибо Майклу Стивенсу (Michael Stephens), который вообще предложил мне заняться этим делом, – проект оказался очень интересным. Многие сотрудники издательства трудились над редактированием, корректурой, подготовкой иллюстраций и выполнением кучи другой работы, связанной с производством книги, – спасибо всем вам.

Особо хочется отметить двух сотрудников Manning. Во-первых, спасибо Берту Бейтсу (Bert Bates) за то, что еще на ранних этапах работы он показал мне, как из сухого справочного материала сделать текст, который было бы приятно читать. Когда я начинал писать очередную главу, на периферии сознания неотступно присутствовала мантра: «Объясни Берту, почему ему нужно это знать...» Во-вторых, спасибо редактору Сюзанне Клайн (Susanna Kline), которая следила, чтобы каждая глава строго соответствовала своему названию, не позволяла мне расслабляться и заставляла выдерживать график.

Вокруг Dart сформировалось весьма активное сообщество разработчиков, которое пользуется для общения списком рассылки dartlang и системой Google+. Я хочу поблагодарить его участников Джона Эванса (John Evans) и Кэвина Мура (Kevin Moore) за техническое редактирование, а также Адама Сингера (Adam Singer), Мэтью Батлера (Matthew Butler) и Ладислава Тона (Ladislav Thon), чьи предложения всегда были желанны.

Благодарю также всех членов сообщества, которые читали книгу и давали отзывы на разных этапах ее написания: Андрэ Роберже (André Roberge), Кард Тасвелл (Carl Taswell), Чэд Дэвис (Chad Davis), Крэйг Ланкастер (Craig Lancaster), Дилан Скотт (Dylan Scott), Глен Стокол (Glenn Stokol), Йон Скит (Jon Skeet), Оливье Нугье (Olivier Nouguier), Рик Гофф (Rick Goff), Родни Боллинджер (Rodney Bollinger), Рокеш Дженки (Rokesh Jankie), Стив Притти

(Steve Pretty), Терри Бэрч (Terry Birch) и Вильгельм Леман (Willhelm Lehman).

Также спасибо всем, кто писал в форуме книги, указывая на неизбежные опечатки, и всем читателям предварительных вариантов – участникам программы Manning's Early Access Program (MEAP).

Наконец, спасибо всем разработчикам Dart, в том числе Сету Лэду, который помогал мне и многим другим ранним приверженцам оставаться в курсе изменений по мере эволюции Dart от первоначальной версии к той, что мы имеем сегодня. Отдельная благодарность Сету за любезное согласие написать предисловие к этой книге.

Об этой книге

Данная книга призвана помочь вам изучить язык Dart, понять его экосистему и писать на нем код, работающий в современных браузерах и на сервере. Вы будете использовать последние технологии HTML5, которые позволяют создавать приложения, работающие в браузере без подключения к сети, а также писать на Dart серверный код, поддерживающий двустороннюю связь с браузерами.

Будучи структурным языком, Dart идеален для разработки крупномасштабных приложений силами территориально разнесенной команды. А благодаря наличию инструментов для автоматической проверки кода, написанного всеми членами команды, Dart облегчает жизнь разработчикам.

Предполагаемая аудитория

Эта книга рассчитана на разработчиков, недовольных отсутствием должной языковой структуры и инструментальных средств для создания браузерных приложений. Если вы на рабочем уровне владеете Java, C# или JavaScript, то сможете сразу же приступить к работе с Dart.

И программисты, предпочитающие писать интерактивные пользовательские интерфейсы, и те, кому больше по душе разработка эффективного серверного кода, обнаружат, что Dart в сочетании с современными браузерными технологиями привносит на сторону клиента структуру, характерную для сервера, а на сторону сервера – гибкость, динамичность и скорость разработки, присущую клиентским приложениям.

Эта книга поможет быстро освоить идеи языка Dart как начинающему веб-разработчику, так и поднаторевшему в написании структурного кода. Все новые концепции поясняются на примерах. В тексте отмечаются как черты, роднящие Dart с другими языками, в частности Java и JavaScript, так и различия между ними.

Для Dart, как и для Java, имеются великолепные инструментальные средства. С другой стороны, Dart, как и JavaScript, не требует этапа компиляции, а это значит, что вы сможете очень быстро приступить к созданию клиентских и серверных приложений на Dart.

Структура книги

Эта книга построена так, чтобы читатель как можно скорее начал работать с Dart. Она состоит их четырех частей. Первая часть содержит обзорные главы.

- В главе 1 приводятся общие сведения об идеях и средствах языка и рассказывается о том, почему вообще Dart появился. Вы узнаете о философии, стоящей за Dart, и о том, какого рода крупномасштабные веб-приложения можно разрабатывать с его помощью.
- В главе 2 рассматривается более широкая экосистема, сложившаяся вокруг Dart, в том числе богатейшие инструментальные средства, которые вы получаете, выбирая структурный язык веб-разработки, созданный компанией, являющейся лидером на этом рынке. Обладая необходимыми техническими ресурсами, Google создала не только сам язык, но и IDE, специализированный браузер для быстрой разработки, серверную виртуальную машину и другие инструменты, нужные для создания высококачественного кода.
- В главе 3 мы напишем простенькое веб-приложение, чтобы понять, как Dart взаимодействует с браузером. Мы сконструируем пользовательский интерфейс, будем прослушивать события браузера и создадим автономные тесты, подтверждающие корректность кода.

Во второй части рассматриваются базовые языковые средства.

- Глава 4 посвящена функциям, которые в Dart являются полноценными объектами. Пишущим на JavaScript некоторые приемы функционального программирования покажутся знакомыми, тогда как разработчики на Java и C# найдут много новых для себя идей, типичных для браузерных приложений.
- В главе 5 мы продолжим конструировать приложение, воспользовавшись встроенной в Dart системой библиотек, и покажем, как эта система соотносится с ограничением доступа. Механизм ограничения доступа в Dart, возможно, станет сюрпризом для разработчиков на Java и C# и окажется желанным подарком для программистов на JavaScript.

- В главах 6, 7 и 8 исследуется структура классов и интерфейсов в Dart. Классы образуют костяк любого сколько-нибудь масштабного приложения, поэтому без умения строить эффективные иерархии классов и пользоваться библиотечными классами, написанными другими людьми, никак не обойтись.
- В главе 9 мы вернемся к функциональному программированию и разберемся с асинхронной природой API доступа к веб. Вы узнаете, как работать с будущими значениями, то есть с переменными, которые получат значение в будущем. Это подготовит почву для работы с API, предоставляемыми клиентскими и серверными библиотеками Dart.

В третьей части обсуждается разработка клиентских приложений, работающих в браузере.

- В главе 10 вы узнаете о цикле обработки событий и о создании пользовательских интерфейсов в Dart.
- В главе 11 в структуру приложения добавляются браузерная навигация, сохранение данных на стороне клиента и обработка данных, представленных в формате JSON.
- Добравшись до главы 12, вы уже будете готовы подключить свое приложение к внешним системам, например к внешнему JavaScript-коду и сторонним серверным API. Хотя Dart рассчитан на все современные браузеры, в этой главе мы научимся создавать пакет для развертывания в качестве приложения Chrome в магазине Google Web Store.

В части 4 речь пойдет об интерфейсе с серверным кодом.

- В главе 13 мы напишем на Dart командное приложение, обращающееся к файловой системе и отправляющее данные по протоколу HTTP, то есть разработаем простой файловый сервер.
- В главе 14 мы подключим клиентское приложение к серверной базе данных и организуем двустороннее взаимодействие по технологии Web Sockets для проталкивания данных клиенту.
- В главе 15, уже зная о взаимодействии с сервером, вы сможете понять, как Dart решает проблему параллелизма с помощью системы изоляторов – модели многопоточности на основе передачи сообщений, более безопасной, чем эквивалентные механизмы в Java и C#. Мы также воспользуемся системой изоляторов для динамической загрузки кода на Dart в работающее приложение. Тем самым мы заложим фундамент для разработки расширений и подключаемых модулей.

В приложениях содержится краткое справочное руководство по базовому языку Dart с примерами, иллюстрирующими синтаксические особенности и странности Dart.

Графические выделения и загрузка исходного кода

Весь исходный код в тексте выделяется моноширинным шрифтом. В книге много фрагментов кода и диаграмм, которые представляют собой законченные аннотированные листинги, иллюстрирующие основные идеи. Все это, как правило, тесно связано с окружающим текстом и составляет неотъемлемую часть изучения Dart.

Иногда код приходилось переформатировать, чтобы он поместился на страницу, но в общем случае код уже написан с учетом ограничений по ширине строки. Сами примеры часто упрощены, чтобы продемонстрировать какую-то важную концепцию, не отвлекаясь на детали, но в основном тексте и в аннотациях к коду приводятся дополнительные подробности.

Весь исходный код, представленный в этой книге, можно скачать с сайта издательства по адресу www.manning.com/DartinAction.

Требования к программному обеспечению

Для работы с Dart требуется как минимум Dart SDK, который можно скачать с сайта www.dartlang.org. В состав Dart SDK включены редактор Dart Editor, специализированный браузер Dartium (необходимый для быстрой разработки на Dart) и конвертер Dart в JavaScript. Имеются версии для Windows, Mac и Linux.

Автор в сети

Приобретение книги «Dart в действии» открывает бесплатный доступ к закрытому форуму, организованному издательством Manning Publications, где вы можете оставить свои комментарии к книге, задать технические вопросы и получить помощь от автора и других пользователей. Получить доступ к форуму и подписаться на список рассылки можно на странице www.manning.com/DartinAction. Там же написано, как зайти на форум после регистрации, на какую помощь можно рассчитывать, и изложены правила поведения в форуме.

Издательство Manning обязуется предоставлять читателям площадку для общения с другими читателями и автором. Однако это не означает, что автор обязан как-то участвовать в обсуждениях; его